Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

# Self-certified multi-proxy signature schemes with message recovery*

Tzong-sun WU[1], Chien-lung HSU[†‡2], Han-yu LIN[3]

([1]*Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung 202, Taiwan, China*)

([2]*Department of Information Management, Chang Gung University, Taoyuan 333, Taiwan, China*)

([3]*Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan, China*)

[†]E-mail: clhsu@mail.cgu.edu.tw

Received Apr. 19, 2008; Revision accepted June 21, 2008; Crosschecked Dec. 22, 2008

**Abstract:** Multi-proxy signature schemes allow the original signer to delegate his/her signing power to *n* proxy signers such that all proxy signers must corporately generate a valid proxy signature on behalf of the original signer. We first propose a multi-proxy signature scheme based on discrete logarithms and then adapt it to the elliptic curve cryptosystem. With the integration of self-certified public-key systems and the message recovery signature schemes, our proposed schemes have the following advantages: (1) They do not require the signing message to be transmitted, since the verifier can recover it from the signature; (2) The authentication of the public keys, verification of the signature, and recovery of the message can be simultaneously carried out in a single logical step; (3) No certificate is needed for validating the public keys. Further, the elliptic curve variant with short key lengths especially suits the cryptographic applications with limited computing power and storage space, e.g., smart cards. As compared with the previous work that was implemented with the certificate-based public-key systems, the proposed schemes give better performance in terms of communication bandwidth and computation efforts.

**Key words:** Self-certified, Multi-proxy signature, Message recovery, Smart cards, Discrete logarithms, Elliptic curve
**doi:**10.1631/jzus.A0820202 **Document code:** A **CLC number:** TN918; TP309

## INTRODUCTION

Since Diffie and Hellman (1976) first proposed the public-key cryptosystem for solving the problem of key management, public-key systems have been widely used in various applications such as e-cash, e-market, and e-voting. In a public-key cryptosystem, each one has a private key and a corresponding public key. To achieve the security requirements of authentication, data integrity and non-repudiation, one can use his/her own private key to generate a digital signature for the given message by using a digital signature scheme. The digital signature will be verified by the signer's public key. Extending the concept of digital signatures, Mambo *et al*.(1996a; 1996b) first introduced the concept of proxy signatures. In a proxy signature scheme, an authorized person, called the proxy signer, is delegated from the original signer to generate a proxy signature on behalf of the original signer.

So far, there are four different sorts of delegations: full delegation, partial delegation, delegation by warrant, and partial delegation with warrant. In the full delegation (Mambo *et al*., 1996a; 1996b), the proxy signer's signing key is the same as the original signer's private key so that all (proxy) signatures are generated with the same private key. Consequently, it cannot offer secure mechanisms to protect the original signer or the proxy signer from being framed by the other. In the partial delegation (Mambo *et al*., 1996a; 1996b), the proxy signature key is computed from the original signer's private key, while the latter

‡ Corresponding author

* Project (No. 94-2213-E-182-019) supported by the National Science Council, Taiwan, China

cannot be derived from the former. However, it is hard to identify the actual signer of a given signature, since a malicious original signer can impersonate the proxy signer to forge a valid proxy signature. In the delegation by warrant (Varadharajan *et al.*, 1991; Neuman, 1993), the original signer prepares the warrant that contains some necessary proxy information, and then sends it to the proxy signer as the delegation authorization. It requires extra efforts to certify and transmit the warrant. Partial delegation with warrant (Kim *et al.*, 1997) preserves the merits of partial delegation and delegation by warrant. It is computationally infeasible for the proxy signer to derive the original signer's private key from the proxy signature key. Certification of the warrant and validation of the signature can be simultaneously carried out in a single step. Obviously, the fourth approach, partial delegation with warrant, is more flexible and secure as compared with the other three, and thus is adopted to implement our proposed schemes.

Generally a secure proxy signature scheme satisfies the following two properties (Mambo *et al.*, 1996a): (1) Unforgeability—No one but the designated proxy signer can generate a valid proxy signature; (2) Verifiability—Any receiver of the proxy signature can verify its legitimacy.

Since we adopt partial delegation with warrant to implement our proposed schemes, the following properties should also be considered (Mambo *et al.*, 1996a): (1) Proxy signer's deviation—The proxy signer cannot create a valid proxy signature with respect to another proxy signer; (2) Distinguishability—A valid proxy signature is distinguishable from a valid original signature; (3) Identifiability—An original signer can identify the actual proxy signer from a given proxy signature; (4) Secret-keys' dependence—The proxy signer's secret key is computed from the original signer's secret key; (5) Undeniability—The proxy signer cannot deny his/her signatures. Up to now, lots of variations of proxy signatures have been proposed (Kim *et al*., 1997; Lee *et al*., 1998; Sun *et al*., 1999; Hwang and Shi, 2000; Hwang *et al*., 2000; Yi *et al*., 2000; Hsu *et al*., 2001; Hwang and Chen, 2001; Lin *et al*., 2002; Tzeng *et al*., 2004; Xue and Cao, 2004a; 2004b).

In a multi-proxy signature, the original signer delegates his signing power to two or more proxy signers, and all of the proxy signers must cooperatively sign on behalf of the original signer. This paper gives a solution to the most common problem of the delegation in enterprise management. For example, two or more vice presidents can corporately make a significant decision or sign an important document on behalf of the president in his absence.

Consider the scenario that a malicious adversary may plot the impersonation attack by substituting the fake public key for the genuine one (Michels and Horster, 1996). In this case, it is necessary to authenticate the public key before using it. A certificate-based public-key system (Kohnfelder, 1978; ISO/IEC 9798-3, 1993; ISO/IEC 14888-3, 1998) is a commonly used solution, in which each public key is accompanied with a certificate issued by the certification authority (CA). One can perform certificate verification to make sure the authenticity of the received public key before using it. It is obvious that extra computation efforts and communication overheads are required for verifying and transmitting the certificate. Shamir (1984) introduced the concept of ID-based public-key cryptosystems, where the public key is defined as the identifier of the user. Obviously, the authenticity of the public key can be explicitly verified without any extra certificate, since the public identifier of each user is intrinsically known. The security of the system heavily relies on the system authority, since all private keys are generated by the system authority. Girault (1991) introduced the concept of a self-certified public-key system. The user can determine his private key, while the public key of the user is generated by CA. In the subsequent signature verification or other cryptographic applications, the verification of the signature and the authentication of the public keys are simultaneously carried out in a single logical step. As compared with certificate-based systems, this approach greatly reduces the computation efforts and communication overheads, since no additional certificates are required. From the above discussions, one can see that the self-certified public-key system might be a better choice for implementing secure and efficient cryptographic applications.

The main advantage of the message recovery signature scheme is that the message can be recovered from the received signature. Hence, the message is unnecessary to be transmitted along with the signature, which results in more bandwidth savings. In this

paper, we adopt the merits of the self-certified public-key systems and the message recovery signature scheme to implement efficient multi-proxy signature schemes. A significant feature of our proposed schemes is that the tasks of verifying the signature, authenticating the public keys and recovering the message are simultaneously carried out within one step. That is, the proposed schemes can improve the efficiency of communication and computation.

Elliptic curve cryptography (ECC) (ANSI X9.31, 1998; ANSI X9.62, 1998; ISO/IEC 14888-3, 1998; IEEE P1363, 2000; ANSI X9.63, 2001; ISO/IEC 15946-3, 2002), first introduced by Miller (1985) and Koblitz (1987), especially suits the applications with limited computing power and insufficient storage space, such as smart cards. To achieve the same level of security as conventional cryptography, ECC requires shorter key lengths, which ensure faster execution and more bandwidth savings (Jurisic and Menezes, 1997; Stallings, 2002). In this paper, we further present an elliptic curve variant of our proposed multi-proxy signature scheme.

The remainder of this paper is organized as follows. In Section 2, we propose a multi-proxy signature scheme based on the discrete logarithms over a finite field. An elliptic curve variant based on the elliptic curve discrete logarithms is given in Section 3. Section 4 discusses some security considerations and the performance evaluation. Also, we compare our proposed scheme with two existing multi-proxy signature schemes. Finally, conclusions are given in Section 5.

## PROXY SIGNATURE SCHEME BASED ON DISCRETE LOGARITHMS

In this section, we propose a multi-proxy signature scheme over a finite field. In the system, there exists a trusted CA whose tasks are to set up the system and to generate users' private and public-key pairs in the registration stage. Initially, the CA determines the following parameters: $p$, $q$—two large primes, and $q|p-1$; $g$—a generator with order $q$ over $GF(p)$; $h(\cdot)$—a secure one-way hash function that accepts input of any length and generates a fixed-length output; $(\delta, \beta)$—the CA's private and public keys, where $\delta \in_R \mathbb{Z}_q^*$ (In this paper, we denote "$\in_R$" as "randomly chosen from"), and

$$\beta = g^{\delta} \bmod p. \tag{1}$$

The parameters $p$, $q$, $g$, $\beta$ and the hash function $h$ are made public, while the CA's private key $\delta$ is kept secret. The proposed scheme consists of three stages: registration, proxy share generation, and multi-proxy signature generation and verification. Details of each stage are described below.

1. Registration stage

Each user $U_i$ associated with the identifier $ID_i$ performs the following interactive steps with the CA:

Step 1: $U_i$ chooses an integer $t_i \in_R \mathbb{Z}_q^*$, computes

$$v_i = g^{h(t_i \| ID_i)} \bmod p, \tag{2}$$

and sends $(v_i, ID_i)$ to the CA, where "||" is the concatenation symbol.

Step 2: After receiving $(v_i, ID_i)$, the CA chooses $a_i \in_R \mathbb{Z}_q^*$, computes

$$y_i = v_i h(ID_i)^{-1} g^{a_i} \bmod p, \tag{3}$$

$$w_i = a_i + \delta h(y_i \| ID_i) \bmod q, \tag{4}$$

and returns $(y_i, w_i)$ to $U_i$.

Step 3: $U_i$ first computes

$$x_i = w_i + h(t_i \| ID_i) \bmod q, \tag{5}$$

and verifies its validity by checking if

$$\beta^{h(y_i \| ID_i)} h(ID_i) y_i = g^{x_i} \pmod p. \tag{6}$$

If Eq.(6) holds, user $U_i$ accepts $(x_i, y_i)$ as his/her private and public keys.

Note that Eq.(6) also validates the authenticity of $y_i$ with respect to $x_i$. Accordingly, there is no need to transmit any certificate along with the public key $y_i$. Eq.(6) can be easily verified as follows:

$$\begin{aligned}
\beta^{h(y_i \| ID_i)} h(ID_i) y_i &= \beta^{h(y_i \| ID_i)} v_i g^{a_i} && \text{by Eq.(3)} \\
&= v_i g^{a_i + \delta h(y_i \| ID_i)} && \text{by Eq.(1)} \\
&= g^{h(t_i \| ID_i)} g^{a_i + \delta h(y_i \| ID_i)} && \text{by Eq.(2)} \\
&= g^{h(t_i \| ID_i) + w_i} && \text{by Eq.(4)} \\
&= g^{x_i} \pmod p. && \text{by Eq.(5)}
\end{aligned}$$

2. Proxy share generation stage

Let $G=\{U_{p_1}, U_{p_2}, ..., U_{p_n}\}$ be the set of $n$ designated proxy signers and $U_0$ the original signer who wants to delegate his/her signing power to the designated proxy group $G$. $U_0$ distributes proxy shares to the members in $G$ with the following steps:

Step 1: $U_0$ chooses an integer $k_i \in_R \mathbb{Z}_q^*$ (for $i$=1, 2, …, $n$) and computes

$$K_i = g^{k_i} \bmod p, \tag{7}$$

$$K = \prod_{i=1}^{n} K_i \bmod p, \tag{8}$$

$$\sigma_i = x_0 n^{-1} + k_i h(m_w \| K) \bmod q, \tag{9}$$

where $m_w$ is the warrant consisting of the original and the proxy signers' identifiers, the delegation duration, and so on.

Step 2: $U_0$ sends ($\sigma_i$, $m_w$) to $U_{p_i} \in G$ via a secure channel and broadcasts ($K_i$, $K$).

Step 3: $U_{p_i} \in G$ verifies the validity of ($\sigma_i$, $m_w$) by checking that

$$g^{\sigma_i} = (\beta^{h(y_0\|ID_0)} h(ID_0) y_0)^{n^{-1}} K_i^{h(m_w\|K)} \pmod p. \tag{10}$$

The correctness of Eq.(10) is shown as follows. By raising both sides of Eq.(9) to exponent with base $g$, we have

$$\begin{aligned} g^{\sigma_i} &= g^{x_0 n^{-1} + k_i h(m_w\|K)} = g^{x_0 n^{-1}} K_i^{h(m_w\|K)} && \text{by Eq.(7)} \\ &= (\beta^{h(y_0\|ID_0)} h(ID_0) y_0)^{n^{-1}} K_i^{h(m_w\|K)} \pmod p. && \text{by Eq.(6)} \end{aligned}$$

3. Multi-proxy signature generation and verification stage

For signing $m$ on behalf of the original signer $U_0$, each $U_{p_i} \in G$ performs the following steps:

Step 1: Choose an integer $z_i \in_R \mathbb{Z}_q^*$, compute

$$r_i = g^{z_i} \bmod p, \tag{11}$$

and broadcast $r_i$ to all other proxy signers.

Step 2: After receiving all $r_j$'s ($j$=1, 2, ..., $n$; $j \neq i$) from other proxy signers, compute $r$ and $s_i$ with the following equations:

$$r = (m \| h(m)) \prod_{j=1}^{n} r_j^{r_j} \bmod p, \tag{12}$$

$$s_i = z_i r_i + (\sigma_i + x_i) h(r \| K) \bmod q, \tag{13}$$

and then broadcast $s_i$ to all other proxy signers.

Step 3: Validate ($r_j$, $s_j$) sent from the proxy signer $U_{p_j} \in G$ ($j \neq i$) by checking that

$$\begin{aligned} g^{s_j} = r_j^{r_j} \Big( & \beta^{n^{-1} h(y_0\|ID_0) + h(y_j\|ID_j)} h(ID_0)^{n^{-1}} \\ & \cdot h(ID_j) y_0^{n^{-1}} y_j K_j^{h(m_w\|K)} \Big)^{h(r\|K)} \pmod p. \end{aligned} \tag{14}$$

If Eq.(14) holds, proceed to the next step; otherwise, $s_j$ is requested to be sent again.

Step 4: Compute $s$ with all collected $s_j$'s ($j$=1, 2, ..., $n$) as

$$s = \sum_{j=1}^{n} s_j \bmod q. \tag{15}$$

The multi-proxy signature of $m$ is ($K$, $r$, $s$, $m_w$). Note that the message $m$ is unnecessary to be transmitted, since it can be recovered from the signature. To check the validity of the multi-proxy signature, the verifier first computes

$$\begin{aligned} & m \| h(m) \\ & = r g^{-s} \left( \left( \beta^{\sum_{i=0}^{n} h(y_i\|ID_i)} \prod_{i=0}^{n} h(ID_i) y_i \right) K^{h(m_w\|K)} \right)^{h(r\|K)} \pmod p. \end{aligned} \tag{16}$$

With the recovered $m$ and $h(m)$, along with the public one-way function $h$, the verifier can check the integrity of the message and the validity of the multi-proxy signature as well. Since the public keys $y_0$ and $y_i$'s are combined into Eq.(16), no additional public-key verification is needed before multi-proxy signature verification. Thus, the message recovery procedure, the authentication of the public keys, and the validation of the multi-proxy signature are simultaneously carried out in one single equation, Eq.(16).

**Theorem 1** If an individual proxy signature ($r_i$, $s_i$) is properly generated by $U_{p_i} \in G$ with Steps 1 and 2 of the multi-proxy signature generation stage, then it satisfies Eq.(14) (See Appendix for the proof).

**Theorem 2** The verifier can perform Eq.(16) to recover the message $m$ from the multi-proxy signature $(K, r, s, m_w)$ and to check the validity of the signature as well as the signers' public keys (See Appendix for the proof).

## MULTI-PROXY SIGNATURE SCHEME BASED ON ELLIPTIC CURVE DISCRETE LOGARITHMS

In this section, we propose the elliptic curve variant based on our first scheme. A significant property of the variant is that the key length required is shorter than that of our first scheme at the same level of security. In general, the elliptic curve cryptography of a 160-bit modulus is almost as secure as our first scheme of a 1024-bit modulus (Jurisic and Menezes, 1997; Stallings, 2002). Furthermore, the computational efforts required for ECC and for the traditional public-key cryptosystems are considered comparable for equal key lengths. Hence, the elliptic curve variant is more efficient in terms of computational complexities and communication overheads, which especially appeals to smart card applications with limited computing power and insufficient storage space.

In the system initialization, the CA determines the following parameters: $p$—a large prime; $a$, $b$—two parameters for $\mathbb{Z}_p$ satisfying $4a^3+27b^2 \bmod p\neq 0$; $E_p(a, b)$—an elliptic curve $y^2=x^3+ax+b \pmod p$; $q$—a large prime such that $q$ is a divisor of the number of points on the elliptic curve $E_p(a, b)$; $O$—a point at infinity over $E_p(a, b)$; $Q$—the base point of order $q$ over $E_p(a, b)$; $h(\cdot)$—a secure one-way hash function that accepts input of various lengths and generates output of a fixed length (note that input of a point over $E_p(a, b)$ means input of the concatenation of the $x$- and $y$-coordinate of that point); $(\delta, B)$—the CA's private and public keys, where $\delta_i \in_R \mathbb{Z}_q^*$ and

$$B=\delta Q. \tag{17}$$

The parameters $p$, $q$, $O$, $Q$, $B$, the hash function $h$, and the elliptic curve $E_p(a, b)$ are made public, while the CA's private key $\delta$ is kept secret. In the following, all elliptic curve point operations are manipulated over $E_p(a, b)$. We further denote $x(P)$ as the $x$-coordinate of point $P$. The proposed ECC variant also consists of three stages like those of the proposed first scheme. Details of each stage are stated below.

1. Registration stage

Each user $U_i$ associated with the identifier $ID_i$ performs the following interactive steps with the CA.

Step 1: $U_i$ chooses an integer $t_i \in_R \mathbb{Z}_q^*$, computes

$$V_i = h(t_i \,\|\, ID_i)Q, \tag{18}$$

and sends $(V_i, ID_i)$ to the CA.

Step 2: After receiving $(V_i, ID_i)$, the CA chooses $a_i \in_R \mathbb{Z}_q^*$, computes

$$Y_i = h(ID_i)^{-1}(V_i + a_iQ), \tag{19}$$

$$w_i = a_i + \delta h(Y_i \,\|\, ID_i) \bmod q, \tag{20}$$

and returns $(Y_i, w_i)$ to $U_i$.

Step 3: $U_i$ first computes

$$x_i = w_i + h(t_i \,\|\, ID_i) \bmod q, \tag{21}$$

and verifies its validity by checking that

$$h(Y_i \,\|\, ID_i)B + h(ID_i)Y_i = x_iQ. \tag{22}$$

If Eq.(22) holds, user $U_i$ accepts $(x_i, Y_i)$ as his/her private and public keys.

Note that Eq.(22) also validates the authenticity of $Y_i$ with respect to $x_i$. Consequently, it is unnecessary to transmit any extra certificate with the public key $Y_i$. The correctness of Eq.(22) is given as follows:

$$\begin{aligned} &h(Y_i \,\|\, ID_i)B + h(ID_i)Y_i \\ &= h(Y_i \,\|\, ID_i)B + V_i + a_iQ && \text{by Eq.(19)} \\ &= (\delta h(Y_i \,\|\, ID_i) + a_i)Q + V_i && \text{by Eq.(17)} \\ &= w_iQ + V_i && \text{by Eq.(20)} \\ &= (w_i + h(t_i \,\|\, ID_i))Q && \text{by Eq.(18)} \\ &= x_iQ. && \text{by Eq.(21)} \end{aligned}$$

That is, if $(Y_i, W_i)$ is correctly generated by the CA, it will pass the test of Eq.(22).

2. Proxy share generation stage

Let $G = \{U_{p_1}, U_{p_2}, \ldots, U_{p_n}\}$ be the set of $n$ designated proxy signers and $U_0$ the original signer

who wants to delegate his/her signing power to the designated proxy signers $G$. $U_0$ distributes proxy shares to the members in $G$ with the following steps:

Step 1: $U_0$ chooses an integer $k_i \in_R \mathbb{Z}_q^*$ (for $i$=1, 2, …, $n$) and computes

$$K_i = k_i Q, \tag{23}$$

$$K = \sum_{i=1}^{n} K_i, \tag{24}$$

$$\sigma_i = x_0 n^{-1} + k_i h(m_w \| K) \bmod q, \tag{25}$$

where $m_w$ is defined as in the proposed first scheme.

Step 2: $U_0$ broadcasts ($K_i$, $K$) and further sends ($\sigma_i$, $m_w$) to $U_{p_i} \in G$ via a secure channel.

Step 3: $U_{p_i} \in G$ verifies the validity of ($\sigma_i$, $m_w$) by checking that

$$\sigma_i Q = (n^{-1}h(Y_0 \| ID_0))B + (n^{-1}h(ID_0))Y_0 + h(m_w \| K)K_i. \tag{26}$$

The correctness of Eq.(26) is shown as follows. Multiplying both sides of Eq.(25) with the base point $Q$, we have

$$\begin{aligned}\sigma_i Q &= (x_0 n^{-1} + k_i h(m_w \| K))Q \\ &= (x_0 n^{-1})Q + h(m_w \| K)K_i && \text{by Eq.(23)} \\ &= (n^{-1}h(Y_0 \| ID_0))B + (n^{-1}h(ID_0))Y_0 \\ &\quad + h(m_w \| K)K_i. && \text{by Eq.(22)}\end{aligned}$$

Therefore, the test of Eq.(26) is successful on condition that ($K$, $\sigma_i$, $m_w$) is correctly generated by $U_0$.

3. Multi-proxy signature generation and verification stage

For signing $m$ on behalf of the original signer $U_0$, each user $U_{p_i} \in G$ performs the following steps:

Step 1: Choose an integer $z_i \in_R \mathbb{Z}_q^*$, compute

$$R_i = z_i Q, \tag{27}$$

and then broadcast $R_i$ to all other proxy signers.

Step 2: After receiving all $R_j$'s ($j$=1, 2, ..., $n$ and $j \neq i$) from other proxy signers, compute $r$ and $s_i$ with

$$r = (m \| h(m))x\left(\sum_{j=1}^{n} x(R_j)R_j\right)^{-1} \bmod P, \tag{28}$$

$$s_i = z_i(x(R_i)) + (\sigma_i + x_i)h(r \| K) \bmod q, \tag{29}$$

and then broadcast $s_i$ to all other proxy signers.

Step 3: Validate ($R_j$, $s_j$) sent from the proxy signer $U_{p_j} \in G$ ($j \neq i$) by checking that

$$\begin{aligned}s_j Q = x(R_j)R_j + h(r \| K)[(n^{-1}h(Y_0 \| ID_0) + h(Y_j \| ID_j))B \\ + n^{-1}h(ID_0)Y_0 + h(ID_j)Y_j + h(m_w \| K)K_j].\end{aligned} \tag{30}$$

If Eq.(30) holds, proceed to the next step; otherwise, $s_j$ is requested to be sent again.

Step 4: Compute $s$ with all collected $s_j$'s ($j$=1, 2, ..., $n$) as

$$s = \sum_{j=1}^{n} s_j \bmod q. \tag{31}$$

The multi-proxy signature of $m$ is ($K$, $r$, $s$, $m_w$). Note that the signing message is unnecessary to be transmitted, since any verifier can recover it from the signature. To check the validity of the signature, the verifier first computes

$$\begin{aligned}m \| h(m) = rx\Bigg(sQ - \sum_{i=0}^{n} h(r \| K)\big(h(Y_i \| ID_i)B + h(ID_i)Y_i\big) \\ - h(m_w \| K)h(r \| K)K\Bigg) \bmod p.\end{aligned} \tag{32}$$

Then one can use the public one-way function $h$ to ensure the integrity of the message and the validity of the multi-proxy signature. Since the public keys $Y_0$ and $Y_i$'s are combined into Eq.(32), additional public-key verifications are not required before the multi-proxy signature verification. Thus the authentication of the public keys, the validation of the multi-proxy signature, and the message recovery procedure are simultaneously carried out in a single equation, Eq.(32).

**Theorem 3** ($R_i$, $s_i$) that can be correctly generated by Steps 1 and 2 of the multi-proxy signature generation stage of the elliptic curve variant satisfies Eq.(30) (See Appendix for the proof).

**Theorem 4** A valid multi-proxy signature ($K$, $r$, $s$, $m_w$) for the message $m$ can be used to recover $m$ with Eq.(32) (See Appendix for the proof).

## SECURITY CONSIDERATIONS AND PERFORMANCE EVALUATION

In this section, we discuss some security considerations and analyze the performance of our proposed schemes.

### Security considerations

The security of the proposed first scheme is primarily based on the cryptographic assumptions of the discrete logarithm problem (DLP) (Diffie and Hellman, 1976; Menezes *et al*., 1997) and the one-way hash function (OHF) (Diffie and Hellman, 1976; Menezes *et al*., 1997). The security of the elliptic curve variant is primarily based on the assumption of the elliptic curve discrete logarithm problem (ECDLP) (Menezes, 1993; Blake *et al*., 1999; IEEE P1363, 2000), instead of the DLP. Since the elliptic curve variant and our first scheme have almost the same structure, we make a merged discussion rather than separate ones. In the following, security considerations are analyzed from three perspectives: the intractability of private keys, the authenticity of public keys and the unforgeability of the multi-proxy signature.

1. Intractability of private keys

Consider the attack that an outsider wants to derive the CA's private key $\delta$ from Eq.(1)/(17). He/She will face the intractability of the DLP/ECDLP assumption. Further, if a malicious member tries to derive $\delta$ from Eq.(4)/(20), he/she has to know $a_i$, a secret number chosen by the CA and under the protection of the DLP/ECDLP assumption in Eq.(3)/(19). Hence, the CA's private key $\delta$ is secure against both inside and outside attacks.

The security of the user's private key $x_i$, computed from Eq.(5)/(21), depends on the random value $h(t_i\|ID_i)$, which is protected by the DLP/ECDLP assumption in Eq.(2)/(18). If any proxy signer attempts to derive the original signer's private key $x_0$ from Eq.(9)/(25), he/she has to know the secret value $k_i$ chosen by $U_0$. However, it is infeasible to obtain $k_i$, since $K_i$ is protected by the DLP/ECDLP assumption in Eq.(7)/(23). In addition, an adversary may try to reveal the original or the proxy signers' private key(s) from some intercepted information, including the ($r_i$, $s_i$), the resulting multi-proxy signature ($K$, $r$, $s$, $m_w$), and the corresponding signing message $m$. Substituting $\sigma_i$ in Eq.(13)/(29) with Eq.(9)/(25), we have

$$s_i = z_i r_i + (x_0 n^{-1} + k_i h(m_w \| K) + x_i) h(r \| K) \bmod q,$$

$$s_i = z_i (x(R_i)) + (x_0 n^{-1} + k_i h(m_w \| K) + x_i) h(r \| K) \bmod q,$$

respectively. It can be seen that, besides the private keys $x_0$ and $x_i$, there are two unknown random values $k_i$ and $z_i$ protected by the DLP/ECDLP assumption in Eq.(7)/(23) and Eq.(11)/(27), respectively. That is to say, it is impossible for the attacker to obtain the private keys $x_0$ and $x_i$.

2. Authenticity of public keys

Note that a valid public key $y_i$ with respect to $x_i$ and $ID_i$ has to satisfy the verification equality, Eq.(6)/(22). A malicious adversary may attempt to forge a valid pair ($ID_{adv}$, $x_{adv}$, $y_{adv}$) satisfying Eq.(6)/(22). The malicious adversary can first arbitrarily choose his/her identifier $ID_{adv}$ and private key $x_{adv}$, and then tries to compute the forged valid public key $y_{adv}$. Obviously, it is infeasible since he/she will face the intractability of the DLP assumption. Similarly, if he/she first fixes ($ID_{adv}$, $y_{adv}$), and tries to obtain $x_{adv}$, the situation is the same as the previous one. What is more, to generate a valid $ID_{adv}$ with the arbitrarily chosen $x_{adv}$ and $y_{adv}$, the adversary will be confronted with the difficulty of the DLP/ECDLP and the OHF assumptions.

3. Unforgeability of multi-proxy signature

To forge a valid multi-proxy signature ($K$, $r$, $s$, $m_w$) passing the test of Eq.(16)/(32), the adversary may first choose a message $m$ along with ($K$, $r$, $m_w$), and then tries to compute $s$ from Eq.(16)/(32). However, the adversary will face the problem of computing discrete logarithms/elliptic curve discrete logarithms and fail to make it under the DLP/ECDLP assumption.

On the other hand, if the adversary first fixes ($K$, $r$, $s$, $m_w$) and then tries to obtain a message $m$ satisfying Eq.(16)/(32), he/she cannot successfully plot the attack unless he/she has the ability to reverse the OHF. Thus, potential forgery attacks cannot succeed in the proposed scheme under the protection of the DLP/ECDLP and the OHF assumptions.

Consider the well-known chosen-ciphertext attack (Bellare *et al*., 1998). An adversary may submit different arbitrarily chosen signatures ($K$, $r$, $s$, $m_w$) to the decryption oracle $D(\cdot)$, which will return the corresponding plaintext to obtain the desired message. However, the probability of obtaining a meaningful plaintext is negligible. Thus, the chosen-ciphertext attack does not work in our proposed schemes.

**Performance evaluation**

In this subsection we will make some comparisons among our proposed first scheme, Lin *et al.*(2002)'s scheme (the LWH scheme for short), and the Xue-Cao scheme (the XC scheme for short) (Xue and Cao, 2004b). These two schemes have structures with partial delegation similar to ours and both seem secure so far. For convenience, we define $T_h$, $T_m$, $T_i$, and $T_e$ as the time for performing an OHF $h$, a modular multiplication computation, a modular inverse computation, and a modular exponentiation computation, respectively.

The time for performing the modular addition is negligible compared to the computation time of performing other operations, and thus is ignored here. In the following comparisons, we assume that all hash functions are implemented with the VSH (Contini *et al*., 2006), an efficient and provable collision-resistant hash function. Contini *et al.*(2006) also claimed that a VSH hash function requires only 4 modular multiplications, i.e., $T_h \approx 4T_m$. $T_i$ and $T_e$ can be further converted into computational units in $T_m$ (Koblitz *et al*., 2000) for facilitating the evaluation of the computational complexities. The relations between $T_i$, $T_e$ and $T_m$ are: $T_i \approx 3T_m$, $T_e \approx 240T_m$. Detailed comparisons of the computational complexities and communication overheads among these three multi-proxy signature schemes are demonstrated by Tables 1 and 2, respectively.

Since our proposed scheme integrates self-certified public-key systems and the message recovery signature schemes, the tasks of authenticating the public keys, verifying the signature and recovering the message can be simultaneously carried out in a single logical step, which helps reduce the computational complexities. Further, it is unnecessary to transmit the signing message since the signing message can be recovered from the signature, which helps with the bandwidth saving. The rough estimation results in Table 1 for the entire scheme show that our proposed scheme outperforms the LWH and the XC schemes by $(1647n+2426)T_m$ and $(915n+2419)T_m$, respectively. Table 2 shows that the total communication cost of our scheme is less than that of the LWH and the XC schemes, both by $(n^2+n-1)|p|+(6n+3)|q|$. Therefore, we conclude that the proposed scheme is better than the LWH and the XC schemes in terms of computation effort and communication cost.

**Table 2 Comparison of communication overheads**

| Phase | Scheme | Communication cost |
|---|---|---|
| Proxy share generation | LWH | $n\|p\|+3n\|q\|^*$ |
| | XC | $n\|p\|+3n\|q\|^*$ |
| | Proposed | $(n+1)\|p\|+2n\|q\|$ |
| Multi-proxy signature generation | LWH | $(n^2+2n)\|p\|+(4n+2)\|q\|^*$ |
| | XC | $(n^2+2n)\|p\|+(4n+2)\|q\|^*$ |
| | Proposed | $n\|p\|+n\|q\|$ |
| Multi-proxy signature verification | LWH | $2\|p\|+(2n+4)\|q\|^*$ |
| | XC | $2\|p\|+(2n+4)\|q\|^*$ |
| | Proposed | $2\|p\|+3\|q\|$ |
| Total cost | LWH | $(n^2+3n+2)\|p\|+(9n+6)\|q\|^*$ |
| | XC | $(n^2+3n+2)\|p\|+(9n+6)\|q\|^*$ |
| | Proposed | $(2n+3)\|p\|+(3n+3)\|q\|$ |

* The cost for transmitting each public-key certificate of the LWH and the XC schemes is implemented with ElGamal signature (ElGamal, 1985), i.e., $2|q|$. LWH: Lin *et al*.(2006)'s scheme; XC: Xue and Cao (2004b)'s scheme

**Table 1 Comparison of computational complexities**

| Phase | Scheme | Time complexity | Rough estimation |
|---|---|---|---|
| Proxy share generation | LWH | $8nT_e+(5n+1)T_m+(n+1)T_i^*$ | $(1928n+4)T_m$ |
| | XC | $7nT_e+3nT_m^*$ | $1683nT_m$ |
| | Proposed | $5nT_e+5nT_m+(n+1)T_i+(3n+1)T_h$ | $(1220n+7)T_m$ |
| Multi-proxy signature generation | LWH | $(n^2+8n+4)T_e+(n^2+7n+2)T_m+T_i+(n+1)T_h^*$ | $(241n^2+1931n+969)T_m$ |
| | XC | $(n^2+6n+4)T_e+(n^2+4n+1)T_m+T_h^*$ | $(241n^2+1444n+965)T_m$ |
| | Proposed | $(n^2+7n-4)T_e+(n^2+5n-5)T_m+(4n-3)T_h$ | $(241n^2+1701n-977)T_m$ |
| Multi-proxy signature verification | LWH | $(3n+6)T_e+(2n+3)T_m+T_h^*$ | $(722n+1447)T_m$ |
| | XC | $(3n+6)T_e+(2n+4)T_m+T_h^*$ | $(722n+1448)T_m$ |
| | Proposed | $4T_e+5nT_m+(2n+2)T_h$ | $(13n+968)T_m$ |
| Total costs | LWH | $(n^2+19n+10)T_e+(n^2+14n+6)T_m+(n+2)T_i+(n+2)T_h^*$ | $(241n^2+4581n+2420)T_m$ |
| | XC | $(n^2+16n+10)T_e+(n^2+9n+5)T_m+2T_h^*$ | $(241n^2+3849n+2413)T_m$ |
| | Proposed | $(n^2+12n)T_e+(n^2+15n-5)T_m+(n+1)T_i+9nT_h$ | $(241n^2+2934n-6)T_m$ |

* The cost for verifying each public-key certificate of the LWH and the XC schemes is implemented with ElGamal signature verification (ElGamal, 1985), i.e., $T_m+3T_e$. LWH: Lin *et al*.(2006)'s scheme; XC: Xue and Cao (2004b)'s scheme

## CONCLUSION

We have proposed two multi-proxy signature schemes mainly based on the DLP and the ECDLP assumptions, respectively. Preserving the merits of self-certified public-key systems and message recovery signature schemes, our proposed schemes are more efficient than the existing schemes, which were implemented with certificate-based public-key systems. The characteristics of our proposed schemes are:

(1) The validation of the multi-proxy signature, the authentication of the public keys, and the message recovery procedure are simultaneously carried out within a single logical step.

(2) No certificate is required for authenticating the public keys, so as to reduce computation effort and communication overhead.

(3) The signing message is unnecessary to be transmitted with the signature, since it can be recovered from the signature, i.e., it gains more bandwidth savings.

As compared with our first scheme, the elliptic curve variant with shorter key lengths is more attractive to the applications with limited computing power and insufficient storage space, like smart cards.

## References

ANSI X9.31, 1998. Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA).

ANSI X9.62, 1998. Public Key Cryptography for the Financial Service Industry—The Elliptic Curve Digital Signature Algorithm (ECDSA). Draft.

ANSI X9.63, 2001. Public Key Cryptography for the Financial Services Industry—Key Agreement and Key Transport Using Elliptic Curve Cryptography.

Bellare, M., Desai, A., Pointcheval, D., Rogaway, P., 1998. Relations among notions of security for public-key encryption schemes. *LNCS*, **1462**:26-45. [doi:10.1007/BFb0055718]

Blake, I., Seroussi, G., Smart, N., 1999. Elliptic Curves in Cryptography. Cambridge University Press, Cambridge, UK. [doi:10.2277/0521653746]

Contini, S., Lenstra, A.K., Steinfeld, R., 2006. VSH, an efficient and provable collision-resistant hash function. *LNCS*, **4004**:165-182. [doi:10.1007/11761679_11]

Diffie, W., Hellman, M., 1976. New directions in cryptography. *IEEE Trans. Inf. Theory*, **22**(6):644-654. [doi:10.1109/TIT.1976.1055638]

ElGamal, T., 1985. A public key cryptosystem and a signature scheme based on discrete logarithms. *LNCS*, **196**:10-18. [doi:10.1007/3-540-39568-7_2]

Girault, M., 1991. Self-certified public keys. *LNCS*, **547**:490-497. [doi:10.1007/3-540-46416-6_42]

Hsu, C.L., Wu, T.S., Wu, T.C., 2001. New nonrepudiable threshold proxy signature scheme with known signers. *J. Syst. Software*, **58**(2):119-124. [doi:10.1016/S0164-1212(01)00032-2]

Hwang, M.S., Lin, I.C., Lu, J.L., 2000. A secure nonrepudiable threshold proxy signature scheme with known signers. *Int. J. Inf.*, **11**(2):1-8.

Hwang, S.J., Chen, C.C., 2001. A New Multi-proxy Multisignature Scheme. National Computer Symp., p.19-26.

Hwang, S.J., Shi, C.H., 2000. A Simple Multi-proxy Signature Scheme. Proc. 10th National Conf. on Information Security, p.134-138.

IEEE P1363, 2000. Standard Specifications for Public Key Cryptography. The Institute of Electrical and Electronics Engineers, Inc., USA.

ISO/IEC 9798-3, 1993. Information Technology—Security Techniques—Entity Authentication Mechanism—Part 3: Entity Authentication Using a Public Key Algorithm. International Organization for Standardization.

ISO/IEC 14888-3, 1998. Information Technology—Security Techniques—Digital Signature with Appendix—Part 3: Certificate-based Mechanisms. International Organization for Standardization.

ISO/IEC 15946-3, 2002. Information Technology—Security Techniques—Cryptographic Techniques Based on Elliptic Curves—Part 3: Key Establishment. International Organization for Standardization.

Jurisic, A., Menezes, A.J., 1997. Elliptic curves and cryptography. *Dr. Dobb's J.*, **22**(4):26-35.

Kim, S., Park, S., Won, D., 1997. Proxy Signatures, Revised. Proc. Int. Conf. on Information and Communications Security. Springer, Berlin, p.223-232.

Koblitz, N., 1987. Elliptic curve cryptosystems. *Math. Comput.*, **48**(177):203-209. [doi:10.2307/2007884]

Koblitz, N., Menezes, A., Vanstone, S., 2000. The state of elliptic curve cryptography. *Des., Codes Crypt.*, **19**(2-3):173-193. [doi:10.1023/A:1008354106356]

Kohnfelder, L.M., 1978. Toward a Practical Public-key Cryptosystem. BS Thesis, Department of Electronic Engineering, Massachusetts Institute of Technology, USA.

Lee, N.Y., Hwang, T., Wang, C.H., 1998. On Zhang's nonrepudiable proxy signature schemes. *LNCS*, **1438**:415-422. [doi:10.1007/BFb0053752]

Lin, C.Y., Wu, T.C., Hwang, J.J., 2002. Multi-proxy Signature Schemes for Partial Delegation with Cheater Identification. The Second Int. Workshop for Asia Public Key Infrastructure. IOS Press, Amsterdam, Netherlands, p.147-152.

Mambo, M., Usuda, K., Okamoto, E., 1996a. Proxy Signature for Delegating Signing Operation. Proc. 3rd ACM Conf. on Computer and Communications Security, p.48-57. [doi:10.1145/238168.238185]

Mambo, M., Usuda, K., Okamoto, E., 1996b. Proxy signatures: delegation of the power to sign messages. *IEICE Trans.*

*Fundam. Electron. Commun. Comput. Sci.*, **E79-A**(9): 1338-1354.

Menezes, A., 1993. Elliptic Curve Public Key Cryptosystems. Kluwer Academic Publishers, USA.

Menezes, A., Oorschot, P., Vanstone, S., 1997. Handbook of Applied Cryptography. CRC Press, Inc., USA.

Michels, M., Horster, P., 1996. On the risk of disruption in several multiparty signature schemes. *LNCS*, **1163**:334-345. [doi:10.1007/BFB0034859]

Miller, V., 1985. Use of elliptic curves in cryptography. *LNCS*, **218**:417-426. [doi:10.1007/3-540-39799-x_31]

Neuman, B.C., 1993. Proxy-based Authorization and Accounting for Distributed Systems. Proc. 13th Int. Conf. on Distributed Computing Systems, p.283-291. [doi:10.1109/ICDCS.1993.287698]

Shamir, A., 1984. Identity-based cryptosystems and signature schemes. *LNCS*, **196**:47-53. [doi:10.1007/3-540-39568-7_5]

Stallings, W., 2002. Cryptography and Network Security: Principles and Practice. Prentice Hall, Upper Saddle River, NJ.

Sun, H.M., Lee, N.Y., Hwang, T., 1999. Threshold proxy signatures. *IEE Proc.-Comput. Dig. Techn.*, **146**(5): 259-263. [doi:10.1049/ip-cdt:19990647]

Tzeng, S.F., Yang, C.Y., Hwang, M.S., 2004. A nonrepudiable threshold multi-proxy multisignature scheme with shared verification. *Fut. Gen. Comput. Syst.*, **20**(5):887-893. [doi:10.1016/j.future.2004.01.002]

Varadharajan, V., Allen, P., Black, S., 1991. An Analysis of the Proxy Problem in Distributed System. Proc. IEEE Computer Society Symp. on Research in Security and Privacy, p.255-275. [doi:10.1109/RISP.1991.130793]

Xue, Q., Cao, Z., 2004a. A Nonrepudiable Multi-proxy Multisignature Scheme. Joint 1st Workshop on Mobile Future and Symp. on Trends in Communications, p.102-105.

Xue, Q., Cao, Z., 2004b. Improvement of Multi-Proxy Signature Scheme. Fourth Int. Conf. on Computer and Information Technology, p.450-455. [doi:10.1109/CIT.2004.1357236]

Yi, L.J., Bai, G.Q., Xiao, G.Z., 2000. Proxy multi-signature scheme: a new type of proxy signature scheme. *Electron. Lett.*, **36**(6):527-528. [doi:10.1049/el:20000422]

## APPENDIX: PROOFS OF THEOREMS 1~4

**Proof of Theorem 1**

By raising both sides of Eq.(13) to exponent with base $g$, we have

$$g^{s_i} = g^{z_i r_i} g^{(\sigma_i + x_i)h(r\|K)} \pmod{p}.$$

The above equation can be further rewritten as

$$\begin{aligned} g^{s_i} &= r_i^{r_i} g^{(\sigma_i + x_i)h(r\|K)} && \text{by Eq.(11)} \\ &= r_i^{r_i} (\beta^{h(y_i\|ID_i)} h(ID_i) y_i g^{\sigma_i})^{h(r\|K)} && \text{by Eq.(6)} \\ &= r_i^{r_i} \left( \beta^{n^{-1}h(y_0\|ID_0)+h(y_i\|ID_i)} h(ID_0)^{n^{-1}} \right. \\ &\quad \left. \cdot h(ID_i) y_0^{n^{-1}} y_i K_i^{h(m_w\|K)} \right)^{h(r\|K)} \pmod{p}, && \text{by Eq.(10)} \end{aligned}$$

which implies Eq.(14).

**Proof of Theorem 2**

Raising both sides of Eq.(15) to exponent with base $g$ yields

$$\begin{aligned} g^{s} &= g^{\sum_{i=1}^{n} s_i} \\ &= g^{\sum_{i=1}^{n} z_i r_i} g^{\sum_{i=1}^{n} (\sigma_i + x_i) h(r\|K)} && \text{by Eq.(13)} \\ &= g^{\sum_{i=1}^{n} z_i r_i} \left( g^{\sum_{i=1}^{n} \sigma_i} \left( \beta^{\sum_{i=1}^{n} h(y_i\|ID_i)} \prod_{i=1}^{n} h(ID_i) y_i \right) \right)^{h(r\|K)} && \text{by Eq.(6)} \\ &= g^{\sum_{i=1}^{n} z_i r_i} \left( \left( \beta^{\sum_{i=0}^{n} h(y_i\|ID_i)} \prod_{i=0}^{n} h(ID_i) y_i \right) K^{h(m_w\|K)} \right)^{h(r\|K)} \\ &\quad \pmod{p}. && \text{by Eq.(10)} \end{aligned}$$

The above equation can be further rewritten as

$$g^{-\sum_{i=1}^{n} z_i r_i} = g^{-s} \left( \left( \beta^{\sum_{i=0}^{n} h(y_i\|ID_i)} \prod_{i=0}^{n} h(ID_i) y_i \right) K^{h(m_w\|K)} \right)^{h(r\|K)} \pmod{p}.$$

Multiplying the above equation by Eq.(12) and rearranging the result, we have

$$m \,\|\, h(m) = r g^{-s} \left( \left( \beta^{\sum_{i=0}^{n} h(y_i\|ID_i)} \prod_{i=0}^{n} h(ID_i) y_i \right) K^{h(m_w\|K)} \right)^{h(r\|K)} \pmod{p}.$$

Hence, the verifier can correctly recover the message $m$ from Eq.(16). With the assistance of the public OHF, the verifier can check the integrity of the message and the validity of the multi-proxy signature as well. The public keys of the original signer and all proxy signers are also authenticated.

**Proof of Theorem 3**

By multiplying both sides of Eq.(29) with base $Q$, we have

$$\begin{aligned} s_iQ &= z_i(x(R_i))Q + (\sigma_i + x_i)h(r \| K)Q \\ &= x(R_i)R_i + (\sigma_i + x_i)h(r \| K)Q \quad \text{by Eq.(27)} \\ &= x(R_i)R_i + h(r \| K)\{[n^{-1}h(Y_0 \| ID_0) + h(Y_i \| ID_i)]B \\ &\quad + n^{-1}h(ID_0)Y_0 + h(ID_i)Y_i + h(m_w \| K)K_i\} \\ &\quad \text{by Eqs.(22) and (26)} \end{aligned}$$

**Proof of Theorem 4**

Multiplying both sides of Eq.(31) with base $Q$ yields

$$\begin{aligned} sQ &= \sum_{i=1}^{n} s_iQ \\ &= \sum_{i=1}^{n} z_i(x(R_i))Q + \sum_{i=1}^{n}(\sigma_i + x_i)h(r \| K)Q \quad \text{by Eq.(29)} \\ &= \sum_{i=1}^{n} z_i(x(R_i))Q + \sum_{i=1}^{n}\sigma_i h(r \| K)Q \\ &\quad + \sum_{i=1}^{n} h(r \| K)[h(Y_i \| ID_i)B + h(ID_i)Y] \quad \text{by Eq.(22)} \\ &= \sum_{i=1}^{n} z_i(x(R_i))Q + \sum_{i=1}^{n} h(r \| K)[h(Y_i \| ID_i)B \\ &\quad + h(ID_i)Y_i] + h(m_w \| K)h(r \| K)K. \quad \text{by Eq.(26)} \end{aligned}$$

The above equation can be further rewritten as

$$\begin{aligned} \sum_{i=1}^{n} z_i(x(R_i))Q &= \sum_{i=1}^{n}(x(R_i))R_i \\ &= sQ - \sum_{i=0}^{n} h(r \| K)[h(Y_i \| ID_i)B \\ &\quad + h(ID_i)Y_i] - h(m_w \| K)h(r \| K)K. \end{aligned}$$

Combining the above equation with Eq.(28), we have

$$\begin{aligned} m \| h(m) = rx\Bigg( & sQ - \sum_{i=0}^{n} h(r \| K)[h(Y_i \| ID_i)B + h(ID_i)Y_i] \\ & - h(m_w \| K)h(r \| K)K \Bigg) \bmod p. \end{aligned}$$

Hence, the verifier can correctly recover $m\|h(m)$. With the public OHF, the verifier can check not only the integrity of the recovered message but also the validity of the multi-proxy signature.